[0013] Therefore, it is an object of the present invention to provide a method and apparatus for describing object oriented programming at a high level that is easy to use and intuitive in nature.

SUMMARY OF THE INVENTION

[0014] The invention is a method and apparatus for defining, representing, and documenting object oriented programming applications, and particularly, those developed to work in a graphical user interface. The invention represents the applications in one or more diagrams that we term Objects and Events Diagrams (OEDs). The invention allows an application architect to communicate the program idea to the programmer by defining the program basis and the program logic without concern about programming-level details of the program, such as how actions will be executed or in which event actions will take place (which will be determined by the programmer). The OEDs may be used as a program specification or a program requirement description. Alternately, they can be used as a documentation tool to document the logic of the objects and events used in a program.

[0015] In the present invention, the application architect represents a program or program portion with an OED using a plurality of different types of symbols that represent different types of objects, as well as a few additional informative symbols that represent some other significant program elements such as data transfer, inheritance, remote links, and databases. The OED also interconnects the various symbols to show their relationships to each other. The architect can place text within the various symbols or in separate documentation to define attributes of the objects and/or other specific details of the object represented by the symbol.

[0016] The invention can be practiced manually (e.g., with pencil and paper). Alternately, the invention can be embodied in software. In such software, the user is presented with a graphical user interface within which the user can drag and drop the various symbols from one or more menus or pallets onto a work area and interconnect them and fill in the object attribute and other data textually and/or through other GUI tools.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] **FIG. 1** shows the various symbols that are available for use in an objects and events diagram in accordance with one particular embodiment of the invention.

[0018] **FIG. 2** is an exemplary graphical user interface application comprising two windows that can be defined, represented, and/or documented by an objects and events diagram in accordance with the present invention.

[0019] **FIG. 3** is a portion of the graphical user interface shown in **FIG. 2** corresponding to the first window of the two windows.

[0020] **FIG. 4** is a portion of the graphical user interface shown in **FIG. 2** corresponding to the second of the two windows.

[0021] **FIG. 5** is an objects and events diagram in accordance with the present invention documenting the first window shown in **FIGS. 2 and 3**.

[0022] **FIG. 6** is an objects and events diagram in accordance with the present invention documenting the second window shown in **FIGS. 2 and 4**.

[0023] **FIG. 7** is an objects and events diagram in accordance with the present invention documenting in more detail the flights table object shown in **FIG. 6**.

[0024] **FIG. 8** is a system-level objects and events diagram documenting the overall website of which the web page represented in **FIG. 2** is a part.

DETAILED DESCRIPTION OF THE INVENTION

[0025] The present invention is a method, process and/or product that aids in the definition, representation and documentation of object oriented programming applications, particularly, those developed to work in a graphical user interface. The invention represents an object oriented programming language application in one or more diagrams that we term objects and events diagrams or OEDs. The invention is not intended to be limited to any particular embodiment or execution paradigm. For instance, the invention and, particularly, OED diagrams can be prepared by hand by an application architect. Alternatively, the invention can be implemented through a computer program that presents a GUI to the application architect through which he can build an OED, such as by dragging and dropping elements from one or more pallets. Even further, an invention can be incorporated as part of a 5GL programming language in which the user creates an OED through a software GUI and the 5GL program generates code from the OED.

[0026] In accordance with the invention, a plurality of standardized symbols, each different symbol representing a different type of object or other programming element, is available for use. In some embodiments of the invention, the library of symbols can be made extensible by the user, i.e., the user can add his or her own symbols. Further, a set of rules define how the symbols are to be used to develop OEDs. In addition, there are recommendations for using the symbols to create an OED. Recommendations generally are instructions as to how to use the symbols in an OED, but, unlike rules, do not require strict adherence. Described below are one or more particular embodiments of the invention. It is to be understood that any rule discussed below could be merely a recommendation in another embodiment and vice versa (i.e., any recommendation may be a rule in another embodiment).

[0027] **FIG. 1** illustrates the object symbols in accordance with one particular embodiment of the invention. Each symbol represents a different type of object or other significant programming element as will be discussed in greater detail below. However, before describing each of the symbols of this particular embodiment, a general discussion of objects is in order. As discussed above, an object can be almost anything that can be coded into computer instructions. An object may be defined as any data structure with a defined intent that is capable of being represented by a logical sequence of computer commands. Furthermore, all objects are at least one of the following four types:

[0028] 1. an object defined within another object;

[0029] 2. an object assigned to another object;